What is claimed is:

[Claim 1] 1. A method for generating and verifying isolation logic modules in a design of an integrated circuit (IC), the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively checking, for each of the power domain, when an isolation logic module isolating the power domain exists in the design;

verifying the correctness of the isolation logic module existing in the design, when the isolation logic module is identified;

generating an isolation logic module for isolating the power domain, when the power domain is not correctly isolated; and

inserting the generated isolation logic in the design.

[Claim 2] 2. The method of claim 1, wherein the voltage constraints comprise for each power domain: a corresponding wakeup domain, a wakeup/shutdown signal, and a list of steady state values.

[Claim 3] 3. The method of claim 2, wherein each of the steady state values defines a determinable under shutdown conditions.

[Claim 4] 4. The method of claim 2, wherein the wakeup/shutdown signal is generated in the wakeup domain.

[Claim 5] 5. The method of claim 1, wherein the isolation logic module is in a register transfer level (RTL) description.

[Claim 6] 6. The method of claim 1, wherein the design is a RTL description.

[Claim 7] 7. The method of claim 1, wherein the voltage constraints are specified by a user by means of a graphical user interface.

[Claim 8] 8. The method of claim 1, wherein verifying the correctness of the isolation logic module comprises:

simulating shutdown conditions;

comparing each of the output values of the power domain to a respective steady state value, and generating an error report when the comparison results in an inequality;

determining whether at least one isolation cell in the isolation module is not connected to the wakeup/shutdown signal, and generating the error report when the checking results an affirmative answer; and

in response to detecting the wakeup/shutdown signal being generated in the wakeup domain, generating the error report when the checking results a negative answer; otherwise, generating a success report.

[Claim 9] 9. The method of claim 8, wherein the error report comprises at least one of an error type and a cause of the error.

[Claim 10] 10. The method of claim 8, wherein the error report and the success report are displayed to the user.

[Claim 11] 11. The method of claim 8, wherein the shutdown conditions and the output values of the power domain are highlighted in the design by means of a visualization tool.

[Claim 12] 12. The method of claim 8, wherein the isolation cell comprises at least one of an AND gate, an OR gate, and a latch.

[Claim 13] 13. The method of claim 1, wherein generating the isolation logic module comprises producing a description language code implementing the isolation logic module.

[Claim 14] 14. The method of claim 13, wherein the description language comprises at least one of Verilog, VHDL, and a combination of Verilog and VHDL.

[Claim 15] 15. The method of claim 14, wherein the description language code comprises instructions assuring that under shut down conditions each of the output values is equal to a respective steady state value.

[Claim 16] 16. The method of claim 13, wherein inserting the isolation logic module comprises:

instantiating the description language code to form an instance of the isolation logic module;

inserting the instance of the isolation logic module in the wakeup domain;

renaming output names of the power domain; and

assigning the original names of output names of the power domain to outputs of the isolation logic module.

[Claim 17] 17. The method of claim 16, wherein the insertion of the isolation logic is performed using at least one of back referencing analysis and a synthesized netlist.

[Claim 18] 18. The method of claim 17, wherein the synthesized netlist is created using a synthesis tool.

[Claim 19] 19. The method of claim 1, wherein inserting the isolation logic module is preceded by verifying the correctness of the isolation logic module placed in the design.

[Claim 20] 20. The method of claim 1, wherein at least one of a computer aided design (CAD) system, a CAD program, a netlist voltage domain analysis tool, and a RTL voltage domain analysis tool, is used to implement the method.

[Claim 21] 21. A computer program product, comprising a computer-readable medium with instructions to enable a computer to implement a method for generating and verifying isolation logic modules in a design of an integrated circuit (IC), the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively checking, for each of the power domain, whether an isolation logic module isolating the power domain exists in the design;

verifying the correctness of the isolation logic module existing in the design, when the isolation logic module is identified;

generating an isolation logic module for isolating the power domain, when the power domain is not correctly isolated; and

inserting the generated isolation logic in the design.

[Claim 22] 22. The computer program product of claim 21, wherein the voltage constraints comprise, for each power domain: a corresponding wakeup domain, a wakeup/shutdown signal, and a list of steady state values.

[Claim 23] 23. The computer program product of claim 22, wherein each of the steady state values defines a determinable under shutdown conditions.

[Claim 24] 24. The computer program product of claim 22, wherein the wakeup/shutdown signal is generated in the wakeup domain.

[Claim 25] 25. The computer program product of claim 21, wherein the isolation logic module is in a register transfer level (RTL) description.

[Claim 26] 26. The computer program product of claim 21, wherein the design is a RTL description.

[Claim 27] 27. The computer program product of claim 21, wherein the voltage constraints are specified by a user by means of a graphical user interface (GUI).

[Claim 28] 28. The computer program product of claim 21, wherein verifying the correctness of the isolation logic module comprises:

simulating shutdown conditions;

comparing each of the output values of the power domain to a respective steady state value, and generating an error report when the comparison results in an inequality;

checking when at least one isolation cell in the isolation module is not connected to the wakeup/shutdown signal, and generating the error report when the checking results an affirmative answer; and

checking when the wakeup/shutdown signal is generated in the wakeup domain, and generating the error report when the checking results in a negative answer; otherwise, generating a success report.

[Claim 29] 29. The computer program product of claim 28, wherein the error report comprises at least one of an error type and a cause of the error.

[Claim 30] 30. The computer program product of claim 28, wherein the error report and the success report are displayed to the user.

[Claim 31] 31. The computer program product of claim 28, wherein the shutdown conditions and the output values of the power domain are highlighted in the design by means of a visualization tool.

[Claim 32] 32. The computer program product of claim 28, wherein the isolation cell comprises at least one of an AND gate, an OR gate, and a latch.

[Claim 33] 33. The computer program product of claim 21, wherein generating the isolation logic module comprises producing a description language code implementing the isolation logic module.

[Claim 34] 34. The computer program product of claim 23, wherein the description language comprises at least one of Verilog, VHDL, and a combination of Verilog and VHDL.

[Claim 35] 35. The computer program product of claim 24, wherein the description language code comprises instructions assuring that under shutdown conditions each of the output values is equal to a respective steady state value.

[Claim 36] 36. The computer program product of claim 33, wherein inserting the isolation logic module comprises:

instantiating the description language code to form an instance of the isolation logic module;

inserting the instance of the isolation logic module in the wakeup domain;

renaming output names of the power domain; and

assigning the original names of the power domain's output names to outputs of the isolation logic module.

[Claim 37] 37. The computer program product of claim 36, wherein the insertion of the isolation logic is performed using at least one of a back referencing analysis and a synthesized netlist.

[Claim 38] 38. The computer program product of claim 37, wherein the synthesized netlist is created using a synthesis tool.

[Claim 39] 39. The computer program product of claim 21, wherein inserting the isolation logic module is preceded by verifying the correctness of the isolation logic module placed in the design.

[Claim 40] 40. The computer program product of claim 21, wherein at least one of a computer aided design (CAD) system, a CAD program, a netlist voltage domain analysis tool, and a RTL voltage domain analysis tool is used to implement the method.

[Claim 41] 41. A method for generating isolation logic modules in a design of an integrated circuit (IC), the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively producing, for each of the power domain using the voltage constraints, a description language code implementing the isolation logic module;

instantiating the description language code to form an instance of the isolation logic module;

inserting the instance of the isolation logic module in a wakeup domain; renaming output names of the power domain; and

assigning the original names of output names of the power domain to outputs of the isolation logic module.

[Claim 42] 42. The method of claim 41, wherein the voltage constraints include for each power domain: a corresponding wakeup domain, a wakeup/shutdown signal, and a list of steady state values.

[Claim 43] 43. The method of claim 42, wherein the voltage constraints are specified by a user by means of a graphical user interface.

[Claim 44] 44. The method of claim 41, wherein the description language comprises at least one of Verilog, VHDL, or combination of Verilog and VHDL.

[Claim 45] 45. The method of claim 42, wherein the description language code comprises instructions assuring that under shut down conditions each of the output values is equal to a respective steady state value.

[Claim 46] 46. The method of claim 41, wherein the insertion of the isolation logic is performed using at least one of a back referencing analysis and a synthesized netlist.

[Claim 47] 47. The method of claim 46, wherein the synthesized netlist is created using a synthesis tool.

[Claim 48] 48. The method of claim 41, wherein inserting the isolation logic module is preceded by verifying the correctness of the isolation logic module placed in the design.

[Claim 49] 49. The method of claim 41, wherein at least one of a computer aided design (CAD) system, a CAD program, a netlist voltage domain analysis tool, and a RTL voltage domain analysis tool is used to implement the method.

[Claim 50] 50. A computer program product, including a computer-readable medium with instructions to enable a computer to implement a method for generating isolation logic modules in a design of an integrated circuit (IC), the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively producing, for each of the power domain using the voltage constraints, a description language code implementing the isolation logic module;

instantiating the description language code to form an instance of the isolation logic module;

inserting the instance of the isolation logic module in a wakeup domain; renaming output names of the power domain; and

assigning the original names of the power domain's output names to outputs of the isolation logic module.

[Claim 51] 51. The computer program product of claim 50, wherein the voltage constrains include for each power domain: a corresponding wakeup domain, a wakeup/shutdown signal, and a list of steady state values.

[Claim 52] 52. The computer program product of claim 51, wherein the voltage constraints are specified by a user by means of a graphical user interface.

[Claim 53] 53. The computer program product of claim 50, wherein the description language comprises at least one of Verilog, VHDL, and a combination of Verilog and VHDL.

[Claim 54] 54. The computer program product of claim 51, wherein the description language code comprises instructions assuring that under shut down conditions each of the output values is equal to a respective steady state value.

[Claim 55] 55. The computer program product of claim 50, wherein the insertion of the isolation logic is performed using at least one of a back referencing analysis and a synthesized netlist.

[Claim 56] 56. The computer program product of claim 55, wherein the synthesized netlist is created using a synthesis tool.

[Claim 57] 57. The computer program product of claim 50, wherein inserting the isolation logic module is preceded by verifying the correctness of the isolation logic module placed in the design.

[Claim 58] 58. The computer program product of claim 50, wherein at least one of a computer aided design (CAD) system, a CAD program, a netlist voltage domain analysis tool, and a RTL voltage domain analysis tool is used to implement the method.

[Claim 59] 59. A method for verifying the correctness of isolation logic modules in a design of an integrated circuit (IC), the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively simulating shutdown conditions for each of the power domain,

comparing each of the output values of the power domain to a respective steady state value, and generating an error report when the comparison results in an equality;

checking when at least one isolation cell in the isolation module is not connected to a wakeup/shutdown signal, and generating the error report when the checking results in an affirmative answer; and

checking when the wakeup/shutdown signal is generated in a wakeup domain, and generating the error report when the checking results in a negative answer; otherwise, generating a success report.

[Claim 60] 60. The method of claim 59, wherein the wakeup domain, the wakeup/shutdown signal, and the steady state values are part of the voltage constraints.

[Claim 61] 61. The method of claim 59, wherein the error report comprises at least one of an error type, and a cause of the error.

[Claim 62] 62. The method of claim 59, wherein the error report and the success report are displayed to the user.

[Claim 63] 63. The method of claim 59, wherein the shutdown conditions and the output values of the power domain are highlighted in the design by means of a visualization tool.

[Claim 64] 64. The method of claim 59, wherein the isolation cell comprises at least one of an AND gate, an OR gate, and a latch.

[Claim 65] 65. A computer program product, including a computer-readable medium with instructions to enable a computer to implement a method for verifying the correctness of isolation logic modules in a design of an integrated circuit, the method comprising:

specifying a plurality of voltage constraints defining at least one power domain in the design;

iteratively simulating shutdown conditions, for each the power domain;

comparing each of the output values of the power domain to a respective steady state value, and generating an error report when the comparison results in an inequality;

checking when at least one isolation cell in the isolation module is not connected to a wakeup/shutdown signal, and generating the error report when the checking results in an affirmative answer; and

checking when the wakeup/shutdown signal is generated in a wakeup domain, and generating the error report when the checking results in a negative answer; otherwise, generating a success report.

[Claim 66] 66. The computer program product of claim 65, wherein the wakeup domain, the wakeup/shutdown signal, and the steady state values are part of the voltage constraints.

[Claim 67] 67. The computer program product of claim 65, wherein the error report comprises at least one of an error type and a cause of the error.

[Claim 68] 68. The computer program product of claim 65, wherein the error report and the success report are displayed to the user.

[Claim 69] 69. The computer program product of claim 65, wherein the shutdown conditions and the output values of the power domain are highlighted in the design by means of a visualization tool.

[Claim 70] 70. The computer program product of claim 65, wherein the isolation cell comprises at least one of an AND gate, an OR gate, and a latch.

[Claim 71] 71. A system for generating and verifying isolation logic modules in the design of integrated circuit, the system comprising:

a database operable to maintain voltage constraints specified by a user;

a code generator operable to generate description language code of the isolation logic modules;

an insertion unit operable to instantiate and insert in each of the isolation modules a respective wakeup domain;

a checking unit operable to verify the correctness of the isolation logic modules; and

a simulator operable to simulate shutdown conditions.

[Claim 72] 72. The system of claim 71, wherein the insertion unit outputs updated files of the design.

[Claim 73] 73. The system of claim 72, wherein the design files comprise at least one of a register transfer level (RTL) description and a synthesized netlist.

[Claim 74] 74. The system of claim 71, wherein the checking unit generates at least one of an error report and a success report.

[Claim 75] 75. The system of claim 71, further comprising a graphical user interface allowing the user to specify the voltage constraints.